The DOD Software Tech News

October 2001

STN VOL. 4, NO. 4

Software Agents

http://www.iac.dtic.mil/dacs/

Agent Based Computing for Autonomous Intelligent Software

James Hendler, Defense Advanced Research Projects Agency (DARPA)
and Laura Douglass, Schafer Corporation

1. Background

In the complex realm of modern military operations, commanders are dealing with increasingly diverse missions, including operations other than war, expeditionary missions, and controlling dangerous situations in dynamic and uncertain environments. All of these missions are further complicated by the requirement for joint and coalition coordination. Achieving decision superiority in these situations is becoming increasingly difficult and complex. The need to gain the right information at the right time for each type of decision maker in this complex scenario is leading the Department of Defense (DOD) toward distributed information systems that are managed and accessed in a network-centric manner.

A key technological innovation capable of handling the complexity of modern warfare is that of software agents. Agent-based computing focuses on the development of distributed computational entities (software agents) which can act on behalf of, mediate or support the actions of human users and autonomously carry out tasks to achieve goals or assist the activities of the users in achieving those goals. In the military, using these agents will improve our information and decision management capabilities and thus drastically reduce the complexities of modern warfare.

1.1 Defense Advanced Research Projects Agency (DARPA) Agent Based Computing

The Defense Advanced Research Projects Agency (DARPA) has a proven record of developing revolutionary new capabilities in the area of software agents, and continues to take a leadership role in this field. DARPA is currently focusing its research in this area on several initiatives in Agent-Based Computing (ABC). The ABC suite of programs will provide the building blocks for understanding and implementing the Software Agents that are the foundation for intelligent distributed information systems and thus will enhance our ability to reach decision superiority. The three initiatives in this effort are the Control of Agent-Based Systems (CoABS) program, the DARPA Agent Markup Language (DAML) program and the Taskable Agent Software Kit (TASK) program.

Using these programs DARPA will create an environment where intelligent distributed computing will be as easy and ubiquitous in the future as data exchange is now. The payoffs to DOD will be seen in improved interoperability of heterogeneous systems, support for coalition operations, improved intelligence gathering, and more timely command and control.

2. Control of Agent Based Systems (CoABS)

The CoABS program consists of three elements—the agent grid, agent interoperability standards and the scaling of agent control strategies. CoABS has developed novel tools for run-time interoperability among heterogeneous systems and is developing new tools to ensure rapid, real-world system integration with other software agents and entities such as servers, databases, legacy systems and sensors.

CoABS has focused on supporting joint and coalition military operations. In these domains it is necessary to assemble disparate information systems into a coherent interoperating whole without redesigning or reimplementing the systems into a single common architecture. To achieve this, it is necessary to construct an information environment where these systems are accessible to each other and their users, resulting in an intelligent distributed information system.

To achieve an operational environment for such a distributed information system, a mechanism is needed to enable the dynamic, runtime integration of agent, object, and legacy software components into applications. The key to this capability is the CoABS Grid, which is software that provides a service-based middleware infrastructure for managing complex information flows. The Grid supports the development of applications for dynamic domains such as military command and control, which require the composability, adaptability, and autonomy provided by software agents.

Current "legacy" systems can be brought to the grid through software wrappers and service descriptions, allowing their functionality to be tapped without major recoding. In addition, the cooperative nature of the problem solving, using existing software components, allows both military and industrial users to develop large scale applications without large scale software development efforts.

In the next year, CoABS will focus on transitioning those run-time integration capabilities to the military, particularly for command and control applications. Among others, CoABS will be providing C2 infrastructure in the Navy's component for Millennium Challenge 2002, operating in the Agile Commander ATD sponsored by Army CECOM, and contributing to the Air Forces' ATD in effects based operations and several of the Joint Battlespace Infosphere-related projects.

3. DARPA Agent Markup Language (DAML)

Currently, the languages used on the World Wide Web impose a significant limitation on agent based computing. Going far beyond XML, the DAML program's goal is to develop a language that allows information content to be expressed on the web in machine-readable ways. This will make for a whole new generation of Internet capabilities. Prototype tools are being developed to show the potential of such markups to provide revolutionary applications that will change the way humans interact with information.

As reported in PC Week, the World Wide Web Consortium (W3C) has embraced DAML as a step toward what Tim Berners-Lee, the creator of the World Wide Web, calls the "semantic web." DARPA is working with Berners-Lee and the World Wide Web Consortium to ensure DAML fits with its plans for the semantic web and becomes the recommended web language for semantic interoperability. The most up to date version of the language is available on the web, as are documents on its use, tools, and demonstrations (www.daml.org).

An exciting military application for DAML is the Horus initiative, a joint DARPA-Intelink Management Office project, which introduces a new way of organizing and presenting intelligence information to users of Intelink, the DOD network for distributing classified intelligence information. Other early adopters of DAML include the Center for Army Lessons Learned and the Oak Ridge National Laboratory.

4. Taskable Agent Software Kit (TASK)

Developing agent-based software is currently something of a "black art" — an arbitrarily complex process that has many ad hoc methodologies but no significant modeling solutions. Very few rules or protocols have been agreed on for analyzing and modeling complex, large-scale, agent systems that interact with distributed heterogeneous information systems. As a result, DARPA is investing in understanding the principles underlying agent-based computing.

The TASK program's goal is to extend the current scientific and

Continued on page 4

Agent Based Computing for Autonomous Intelligent Software

Continued from page 3

mathematical foundations of agent-based computing and to add precision to the engineering of agent-based systems. TASK researchers will explore the application of mathematics not traditionally used for computer modeling, such as statistical physics and chaotic dynamics, to explore the modeling of multi-agent systems.

TASK researchers will demonstrate their algorithms and approaches against a set of "research exploration frameworks" chosen to be of great relevance to a very wide range of DOD problems. Specific focus will be placed on dynamic command and control problems, the fundamentals of cooperative systems, and understanding information-agent behavior. In this way, new techniques for resource assignment against chaotic problems, hybrid control of dynamic systems, and modeling the information dynamics of complex IT systems will be explored and exploited.

5. Conclusion

In this short article, we have described a vision of the software-of-the-future that drives the agent-based computing programs supported by DARPA, and run in cooperation with the services and other DOD agencies. These programs provide funding to over fifty universities, companies, and research institutes who are cooperating in developing both underlying science and implemented technology for developing and fielding agent based systems for the Department of Defense.

Biographies

Dr. James Hendler is currently working at the Defense Advanced Research Projects Agency (DARPA) as a Program Manager in the Information Technology Office.

Dr. Hendler is on a leave of absence from the University of Maryland where he continues to retain his status as a professor and head of both the Autonomous Mobile Robotics Laboratory and the Advanced Information Technology Laboratory. He has joint appointments in the Department of Computer Science, the Institute for Advanced Computer Studies and the Institute for Systems Research, and he is also an affiliate of the Electrical Engineering Department.

He is the author of the book *Integrating Marker-Passing and Problem Solving: An Activation Spreading Approach to Improved Choice in Planning*. He has also authored over 100 technical papers in artificial intelligence, robotics, intelligent agents and high performance computing. Hendler was the recipient of a 1995 Fulbright Foundation Fellowship, is a member of the US Air Force Science Advisory Board, and is a Fellow of the American Association for Artificial Intelligence.

Ms. Laura Douglass is currently at Schafer Corporation, where she provides technical and analytical support to the Agent Based Computing program at DARPA. Previously, she was the Special Assistant to the Deputy Under Secretary of Defense for Science and Technology, and spent several years providing program management support to the Office of Naval Research while on contract from the Pennsylvania State University's Applied Research Laboratory.

Contact Information

Dr. James Hendler
Program Manager
DARPA / ITO
3701 Fairfax Drive
Arlington, VA 22203-1714

E-mail: jhendler@darpa.mil
Phone: 703-696-2238
Fax: 703-696-2203

Laura Douglass
Sr. Program Manager
Schafer Corporation
4301 North Fairfax Dr., Suite 700
Arlington, VA 22203

E-mail: ldouglass@schafercorp-ballston.com
Phone: 703-516-6033
Fax: 703-516-6065

©Msg*Log: E-mail-based Agent Messaging to Improve Robustness in a Distributed Logistics Planner

Tom Bannon, Steve Ford, Craig Thompson, and David Wells, Object Services and Consulting Inc.

1. Introduction

The Department of Defense (DOD) Joint Vision 2020 [1] identified "focused logistics" as a key component for achieving full spectrum dominance in future conflicts. To that end, the Defense Advanced Research Projects Agency (DARPA) Advanced Logistics Project (ALP) has successfully demonstrated the use of an extensible collection of loosely coupled agents to generate, monitor, and dynamically modify complex, multilevel logistics plans. ALP agents reside on an agent infrastructure called Cougaar (Cognitive Agent Architecture). The current DARPA Ultra*Log program is improving the robustness, scalability, and security of the Cougaar agent infrastructure with the goal of allowing ALP agent-based planning to survive and perform in a highly chaotic environment in which unforeseen or uncontrollable events interfere with the ALP planners' operation, and with the ability of the military procurement, transport, and warehousing organizations to physically execute the plans generated by ALP.

Because the ALP planning process is highly distributed, reliable and timely message delivery to multiple planning agents is mandatory, even as the communications infrastructure degrades and as the planning agents move. The existing Cougaar communications mechanism is based on Java Remote Method Invocation (RMI) extended with message queues and retry policies. This performs well under stable conditions, but rapidly loses its ability to deliver messages as chaos increases due to inherent properties of the RMI protocol. This failure will ultimately make it impossible for Ultra*Log to achieve its stated robustness and performance goals of suffering "no more than 20% capability degradation and 30% performance degradation under conditions of 45% information infrastructure loss in an environment of 90% of maximal real-world chaos".

This article describes the ongoing Msg*Log (Messaging Logistics) effort to improve the robustness of Cougaar agent communications by transparently adding additional message transport mechanisms based on the robust and ubiquitous E-mail and Netnews protocols. The resultant three transport mechanisms have different operational properties that make each suitable under different conditions. To capitalize on this, Msg*Log provides a higher level Adaptive Message Transport capability to automatically switch among transport mechanisms to cope with degradation of the communications infrastructure and to adjust the level of redundancy.

2. The Logistics Problem

Logistics planning and execution is a huge and complex problem, with operational and financial implications for the DOD. The DOD states that:

> *If Desert Shield/Storm logistics had optimized lift scheduling, detailed coordination between planning and execution, and visibility into the logistics pipeline, significant improvements would have been possible in deployment surges and resource sequencing, planning and replanning driven by changing requirements, and improved control over the logistics pipeline.* [11]

Estimates are that this would have allowed the campaign to have concluded 100 days sooner, reduced the quantity of material transported by 1M Tons, and reduced logistics costs by $800M. These represent improvements of approximately 45%, 33%, and 40%, respectively.

Technically, logistics planning and monitoring poses a challenging problem. It is clearly a large problem as evidenced by the Desert Storm example. The types of planning activities are varied, including inventory, packing, route planning, and scheduling, each subject to multitudinous and varied constraints. The problem is highly distributed, yet interconnected; the supported organizations are

Continued on page 6

Msg*Log: E-mail-based Agent Messaging

Continued from page 5

autonomous and can reside anywhere on the globe, but not only does an organization need to know its own part of the overall plan, it needs to know enough about other organizations and their plans to be able to collaborate in planning and execution. Complicating matters further, these organizations can move during the course of an operation, sometimes in an unexpected manner. Additional dynamism enters the system in the form of changed logistics requests and the potential loss of logistics resources as the military operation progresses, any of which may require plans to be modified on-the-fly. The tempo and scope of operations can change rapidly as the situation progresses from peace to deployment to combat to peacekeeping and back to peace. Finally, all this must be accomplished in an environment in which opponents are attempting to prevent logistics activities from being planned and executed by both kinetic and information attacks.

3. Agent-based Logistics Planning

A problem of this nature calls for a highly flexible, scalable, and robust computing solution that can be incrementally improved and expanded. The DARPA Advanced Logistics Project (ALP) and its successor Ultra*Log are developing an agent-based computing solution to distributed logistics under stress.

The following presents sufficient information about ALP and Cougaar to understand the role of Msg*Log. More information about ALP and Ultra*Log can be found at [5] and [6]. Complete details of the Cougaar agent infrastructure on which ALP planning agents reside can be found at [7].

The Cougaar architecture provides the technical basis for a loose confederation of ALP planners that interoperate via message passing. ALP planners, corresponding to Cougaar agents can be grouped into *nodes* and *communities* that perform more complex planning or monitoring. Agents generally perform a very specific task (e.g., job-shop scheduling). A node is Cougaar software that supports one or more agents within a single Java Virtual Machine (JVM); typically several related agents will be placed into a single node to optimize communications (e.g., a scheduler, packing planner, and associated data access supporting a warehousing organization could reside in a single node). A *community* is a logical grouping of agents, typically associated with a DOD organization (e.g., the US Transport Command). Communities can be hierarchical. Communities have no physical interpretation in Cougaar, but since they correspond to an organization, there is often at least some physical "closeness" between members of a community (e.g., on a LAN or within a single base). A *society* is a collection of agents and communities working on a common problem.

All but the simplest plans will involve interaction of multiple agents, node, or communities and will involve considerable remote interaction. Thus, robust ALP planning requires "survivable" internode and intercommunity communication that will continue to operate as networks partition or become swamped as planning agents and nodes migrate along with their owning organization or as part of an overall survivability plan. Figure 1 features a graphic by Dr. Todd Carrico of DARPA, which illustrates some of these concepts.

The current Cougaar implementation uses Java RMI (Java's RPC mechanism) for internode communication. The strength of RMI is its tight integration with Java and its speed. Its weakness is its dependence on point-to-point direct connectivity between sender and receiver and on accurate knowledge of the recipient's IP address and port ID. Under low chaos conditions, these requirements are generally met and when violated are easily remedied, making RMI a good choice, especially when combined with message queues and retry policies (which Cougaar currently supports). However, as chaos increases, these requirements are less likely to be met: partitions become more frequent and take longer to repair, while at the same time, nodes are more likely to migrate. In the worst case, end-to-end connectivity between critical pairs of nodes may never exist, at least not often enough to produce plans in an acceptable time. Thus, even though an RMI-based approach can tolerate highly chaotic conditions, it will not be able to accomplish much during them,

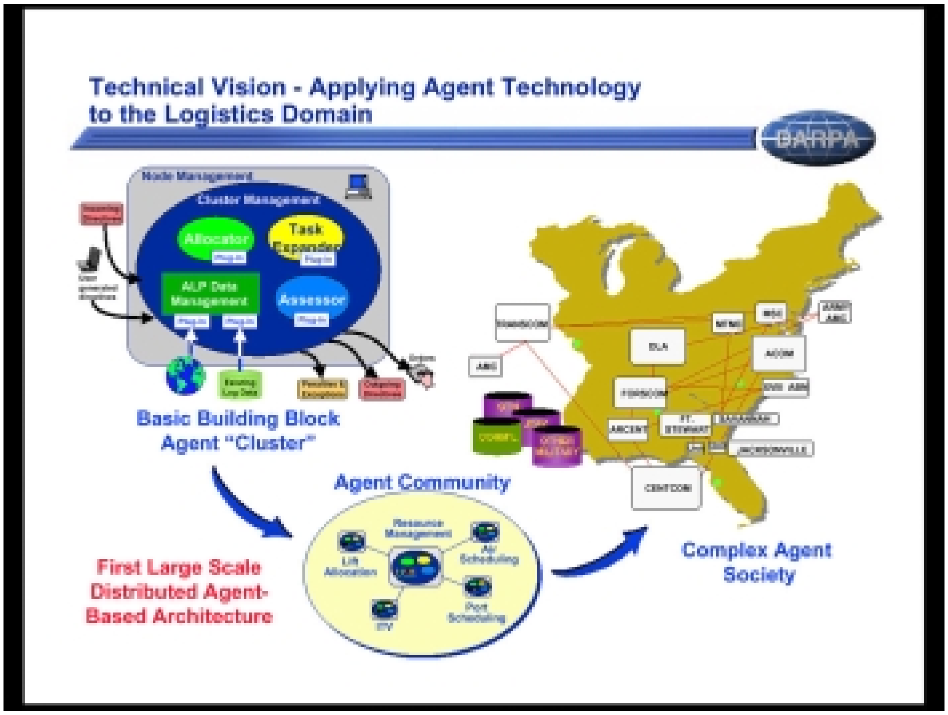

Figure 1: Technical Vision - Applying Agent Technology to the Logistics Domain

which will prevent Ultra*Log from reaching its goal of "no more than 20% capability degradation and 30% performance degradation under conditions of 45% information infrastructure loss in an environment of 90% of maximal real-world chaos".

4. Msg*Log

Clearly, an important requirement for overall ALP/Cougaar survivability [5][6][7] is for inter-agent messaging to be reliable and timely even when the communications infrastructure is stressed. Since Cougaar's RMI-based message transport is not reliable under those conditions, Cougaar needs a better way to perform remote messaging when chaos increases.

Two ways to address this suggest themselves. RMI can be further extended with intermediate servers and naming and redirection mechanisms to provide a store-and-forward capability and support for mobile agents, or alternate transport mechanisms that already provide these capabilities can be adapted for Cougaar use. RMI extensions would require the ability to install software on intermediate network nodes; a degree of access not generally available in third party networks. Since use of such networks is required for many applications, including logistics, RMI extensions alone will be insufficient. Fortunately, robust protocols that allow message delivery across frequently partitioning networks and that support mobile recipients while not requiring control over intermediate nodes already exist, namely Simple Message Transfer Protocol (SMTP) and Network News Transport Protocol (NNTP), the E-mail and UseNet newsgroup protocols.

STMP and NNTP-based internode communications mechanisms have two characteristics that recommend them. Like RMI, SMTP and NNTP are both well defined, mature protocols with many implementations, so any system built on top of them will have industrial-strength underpinnings. The many SMTP and NNTP servers already in place ensure wide "reach" for the ALP system; in fact allowing delivery of messages to places unreachable by RMI. More importantly from a survivability perspective, SMTP and NNTP have different operational characteristics that, while making them slower than RMI, allow them to continue to function with reasonable performance at levels of chaos that would preclude RMI from delivering any messages at all.

Unlike RMI, which requires end-to-end connectivity between sender and receiver, SMTP and NNTP use store-and-forward mechanisms that only require individual hops along a delivery path to be functioning. A consequence is that SMTP and NNTP can deliver messages even if sender and receiver are never connected at any point in time. In a frequently partitioning network environment, this is clearly an advantage.

Continued on page 8

Msg*Log: E-mail-based Agent Messaging

Continued from page 7

Second, RMI requires that a sender have knowledge of its recipient's IP address and port ID, which is difficult to ensure if ALP agents/ nodes and logistics organizations are moving around in cyberspace or the real world. Since in both SMTP and NNTP, messages are "picked up" rather than delivered, all they require is that a sender know how to contact some "mail drop" where a recipient can look for its messages. This allows message delivery in cases where the sender does not know where a mobile recipient resides, but the recipient knows where it can pick up its mail. Since there can be many mail drops and these can be relatively stationary, this provides a much more robust solution. This can be exploited to further improve robustness by sending the same message to multiple servers, making it more likely that a recipient will be able to access its messages from somewhere, even in the face of many failures. NNTP in particular is efficient at distributing messages with high fan out through multiple layers and large numbers of servers, making it particularly useful under highly chaotic conditions.

The new Msg*Log SMTP and NNTP-based transport mechanisms convert Cougaar inter-agent messages into E-mail messages or news postings, transmit them via the existing E-mail or news infrastructure, convert them back to Cougaar messages on the receiving end, and hand them off to the existing Cougaar mechanisms for delivery to the appropriate local agents. For compatibility with the existing Cougaar RMI-based transport, Msg*Log uses Java serialization as an efficient external representation for both SMTP and NNTP message transport. Other encodings are certainly possible; a previous OBJS E-mail-based agent communications system called eGents used XML as en external encoding in the CoABS 24x7 Grid [15] where inter-language portability and human readability was more crucial than performance.

Msg*Log's implementation uses existing E-mail and news servers, and existing Java SMTP, POP3, and NNTP clients. This enables applications to control routing at a level to which they have access, via established Internet services that route public messages between servers out of line with the direct path between two Cougaar nodes. With respect to caching, it should be noted that Internet routers do not cache packets for significant periods of time waiting for a connection to the next router, so an additional advantage of our design over dynamic network routing is that ours operates in the presence of great latency and full disconnectedness.

Cougaar's NameServer is implemented (NameServer is an interface, not a class) to map Cougaar nodes to one or more E-mail addresses. While there are several possibilities for mapping Cougaar agent addressing to E-mail addressing, for compatibility with the Cougaar architecture, our current approach requires an E-mail address for every Cougaar node (effectively a Java VM) in which multiple Cougaar agents can run as threads. Each Cougaar node has at least one E-mail address (an E-mail account on at least one E-mail server used for sending/receiving messages); multiple accounts at multiple servers are possible in the interest of sending and receiving messages via multiple routes. Both the SMTP and NNTP-based transport mechanisms include the ability to pick up E-mail or news-encode inter-agent messages on an appropriate schedule and will have the ability to try alternate sites.

The SMTP and NNTP-based transport mechanisms integrate easily into the Cougaar architecture, which provides explicit support for swapping transport mechanisms by subclassing the Cougaar MessageTransport class. This packaging makes the existence and use of transport alternatives transparent to Cougaar agents and in fact, to the rest of the Cougaar infrastructure. Cougaar agents need not know which transport mechanism is being used. If additional transport mechanisms were needed (e.g., based on broadcast or groupware), they would be added in the same manner.

The ability to route messages via alternate message transports improves the flexibility of internode communications. However, the bigger benefit comes by providing an adaptive mechanism to dynamically choose between those message transports and parameterize their use on a message by message basis using network QoS information and higher level application-dependent intelligence about the relative importance of the message and the likelihood of failure of its delivery for reasons not detectable via QoS metrics. For instance, if direct connectivity is not available between two nodes when a message is sent, then RMI will not work; if the need to send the message immediately is critical, then alternate routes to the recipient could be considered. Knowledge of the network topography combined with the partial information about the failure might suffice to identify a reliable route to the remote node via a chain of SMTP servers, or a set of routes of probable reliability. Alternatively, a security agent might have advised that a route over which direct communication would travel is not to be trusted, and alternate routes should be considered. Finally, general policies could state that under certain operational conditions (an InfoCon), that public networks are not to be used for any data unless encrypted, regardless of the information's inherent sensitivity.

This is the idea behind Msg*Log's Adaptive Message Transport (AMT), a higher level transport mechanism with the ability to select among, parameterize, and monitor the behavior of the various lower-level transport mechanisms known to it. This mechanism merges disparate sources of relevant information, plans, and executes an appropriate communications strategy using the available transport mechanisms. AMT is designed to manage not only messaging via RMI, SMTP, and NNTP, but also to be extensible with future MessageTransport implementations.

5. Metrics

The effectiveness of Msg*Log in improving Cougaar robustness is evaluated by how well it increases the *survivability* of ALP logistics planning capability in the face of stresses to the computing and communications environment. To judge this, we want answers to the following questions:

- When Msg*Log alone is added to Cougaar, how much do logistics plans degrade in quality and timeliness of generation when subjected to various kinds and intensities of communications stress, and how does that compare to identical planning conducted under similar conditions without Msg*Log?
- When all Ultra*Log enhancements are added to Cougaar, how much do logistics plans degrade in quality and timeliness of generation when subjected to various kinds and intensities of communications stress, and how does that compare to identical planning conducted under similar conditions without Ultra*Log enhancements?

Robustness in the face of environmental stresses can only be measured by having a baseline of functionality of the unstressed system. There is no formal notion of a "best possible logistics plan", since it is never possible to state that a given plan could not be improved with more time or better planning tools or smarter logisticians. Thus, an appropriate baseline for planner behavior is: (1) the quality of the logistics plan, (2) generated in some nominal time, (3) by a given ALP society when, (4) requested to satisfy a given set of (possibly changing) logistics requests using (5) a given set of (possibly changing) logistics resources (truck, ports, etc.), when (6) the computing resources (computers, networks, data sets) used in the planning process are not stressed in any way (no failure, degradation, or corruption).

A test measures the effectiveness and operational behavior of Msg*Log across a range of stresses to the computing environment. A test uses a single test case (an ALP society, a computing environment, a set of (time varying) logistics requests, and a set of (time varying) logistics resources) to exercise the system. Testing will be done twice this year, once with a moderate size test case and the other with a 1000 node ALP society. In future years, tests of increasing complexity will be defined by the Ultra*Log program's assessment team. Within a test, several individual experiments inject varying types and intensities of stress. A model for injecting synthetic stresses can be found in [14]. As the computing environment is stressed, it is anticipated that logistics plan quality will degrade. The extent to which logistics plan

Continued on page 10

Continued from page 9

quality does not degrade is a measure of Cougaar robustness. Stresses to be applied in the tests include:

Kinetic attacks

- Permanent loss of geographically adjacent communications links
- Permanent loss of target agents (what will M*L do if the recipient dies?)
- Permanent loss of SMTP and/ or NNTP servers

Information warfare attacks

- Denial of service attacks place excess load on communications links
- Network partitioning
- Rolling network partitions (partition moves across network)
- IW attacks crash SMTP and/or NNTP servers
- Spamming of STMP and/or NNTP servers (causes delays or overflows)
- Protocol attacks on RMI, SMTP, and/or NNTP protocols (renders a transport unusable)
- Loss of stored messages at SMTP and/or NNTP servers

In addition to measuring the effectiveness of Msg*Log in increasing Cougaar robustness, we are interested in collecting operational or infrastructure level statistics about the concrete behavior of Msg*Log under different conditions. These operational statistics are to be used to set Msg*Log policy [3] and to direct future development. For these purposes, we want answers to the following questions about the behavior of the Msg*Log transport mechanisms:

- What is the mechanism's performance under various kinds and intensities of communications stress, and how does that compare with the performance of the original RMI-based transport under similar conditions?
- What is the mechanism's resource utilization under various kinds and intensities of communications stress, and how does that compare with the resource utilization of the original RMI-based transport?
- Under what kinds and intensities of communications stress do each of the Msg*Log transport mechanisms become unable to perform reliable and timely message delivery, and how does this compare to the behavior of the original RMI-based transport?
- Under what kinds and intensities of communications stress does Msg*Log (including the ability to select the "best" transport mechanism) become unable to perform reliable and timely message delivery, and how does this compare to the behavior of the original RMI-based transport?
- How does Msg*Log affect the performance of the original RMI-based transport?

Three kinds of information must be collected to answer the above questions:

- Message delivery information (when the message was sent, the transport mechanism used, message size, delivery time, distance between sender and receiver and the distance the message actually traveled (in terms of LAN and WAN hops), whether the recipient was fixed or mobile (not necessarily known until message is received), and the environmental stresses present.
- Disk and memory utilization at the SMTP & NNTP servers at various time points (for correlation with message send times to create a map of disk usage as a function of number and size of messages)
- Bandwidth utilization

From the raw statistics captured during testing, the following statistics should be computed:

- For each transport mechanism, a scatter plot of message delivery times as a function of distance between sender and receiver (actual distance, not how far the message went)
- For Msg*Log, a scatter plot of message delivery times as a function of distance between sender and receiver (actual distance, not how far the message went)

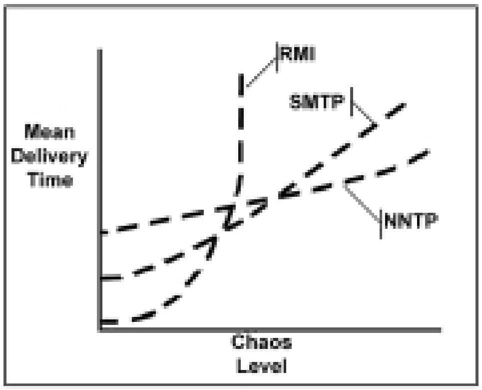

Figure2: Mean Delivery Time as a Function of Environmental Stress (Sample)

- For each of the above, the number of undeliverable messages at each distance (factor out messages that were not delivered due to agent failure, since Msg*Log can't do anything about those)

The different experiments in a test will be conducted at different "levels of stress". The scatter plots produced for the experiments can be sliced to produce scatter plots of the effect of stresses as follows. For each inter-agent distance, a scatter plot is made that maps delivery times (at that inter-agent distance) as a function of the stress level. Figure 2 shows the expected shape of the graphs (read "stress level" for "chaos level; the terms have been used interchangeably in the past, although the term "stress" is preferable as "chaos" has a specific, and different, meaning). Only the mean delivery time is shown, but the actual graphs will be scatter plots.

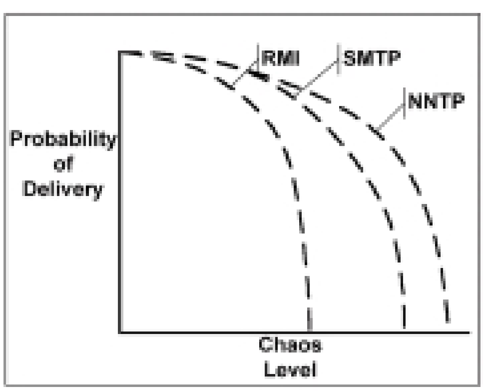

Figure 3: Probability of Message Delivery as a Function of Environmental Stress (Sample)

For all of these, standard statistics such as mean delivery time (excluding undeliverable messages) and fraction of messages delivered in less than a given time T can be computed. If T is chosen meaningfully with regard to the Ultra*Log concept of operations, it becomes a useful predictor of whether a given transport mechanism, or Msg*Log itself, is likely to be able to provide timely message delivery under given stress and distance conditions. The presumed graph of this for a given distance and time T is shown in Figure 3.

6. Status

A Preliminary Msg*Log Design is complete, and implementation of the E-mail and NNTP Message Transports are underway. A proof-of-concept implementation of E-mail Message Transport has been demonstrated using a sample Cougaar society. The first software delivery will include a small number of hand-coded Message Transport policies to select and parameterize the three Message Transports. We expect to integrate with BBN's QoS Monitoring Service [12][13] and evolve the Msg*Log design and implementation in concert with the evolving Cougaar architecture over the life of the program.

7. Plans

In the near term, we will be completing the integration of Msg*Log with the remainder of the Cougaar architecture, participating in survivability testing, doing general performance improvements, and writing some simple policies for Adaptive Message Transport.

Beyond that we will be developing more sophisticated adaptation mechanisms and developing communications policies compatible with evolving higher level survivability and security policies. We will also be investigating other techniques such as stealthy message delivery, traffic analysis masking, and service bidding that could well be supported by message transport schemes that allow delivery to multiple (perhaps intentionally spurious) addresses and delayed message pickup at convenient (and safe) times.

There is a strong possibility that we will add additional message transport mechanisms, in particular one based on extensions to RMI as discussed in Section 4. As noted there, RMI could be extended with intermediate servers, name mapping, and redirection to achieve functionality similar to the SMTP and NNTP transport mechanisms

Continued on page 12

Continued from page 11

discussed in this article. However, the operational behavior might be quite different. In particular, SMTP and NNTP servers are optimized to minimize resources required per message (cpu/message and storage/message), but not necessarily for throughput speed (time/message), although that is generally pretty good as a side effect. An RMI-based transport mechanism using store-and-forward servers built specifically to relay a message as fast as possible might be a useful alternative in proprietary networks where it is possible to install the servers at intermediate nodes.

8. Potential Future Work

There are a number of more speculative avenues of research which we wish to pursue.

We plan to examine the possibility of exercising direct control over message routing at the network level for traffic that can be limited private networks. It is certainly feasible that some logistics communities might have dedicated intranets supporting at least part of their computing assets. We have not chosen this approach initially because over the open Internet today, there is insufficient access to routing information and insufficient control of routing at the network level available to the application to make it work. In this regard, recent work on intelligent swarms [9] in which insect simulations based on simple rules and weighted pheromone trails are applied to networks to find low cost routes that reconfigure when the network is disturbed appear interesting.

Another area of interest is using Msg*Log to support groupware such as ISIS, Horus, and Ensemble [10]. Groupware systems allow the definition of groups of distributed processes and guarantee certain properties of message delivery to the group members, including message ordering, quorum calls, and handling of network partitions and reconnections. This might well be a useful capability for both logistics applications and agent systems in general, since it would allow synchronized communication with groups of agents working on a single task – say logistics “what-iffing”. Groupware systems all rely on message delivery by a lower level transport mechanism; as such Msg*Log is complementary. In the case of Horus, this possibility is enhanced by a replaceable protocol stack and that could be adapted relatively easily to use Msg*Log transport. Interesting technical questions arise when different members of a group must be reached via different transport mechanisms with different delivery properties.

It would also be interesting to explore using the ALP planning capabilities to choose message transport protocols and routing strategies; in effect to treat message delivery itself as a logistics problem occurring at the infrastructure level. A concern is that routing plans could not be made fast enough, with the result that performance would degrade. The severity of this problem would depend on how frequently routing plans would have to change in response to network loads and disruption. It is possible that “normal” selection and routing could be done by Msg*Log policies as described above, with only “hard” problems delegated to the logistics tools.

9. Conclusion

This article has described a system to improve the survivability of an agent infrastructure (Cougaar) that is being used to support a complex logistics planning system. Techniques to measure the improved survivability were also presented.

There has long been speculation that so-called “ilities” (survivability, scalability, quality of service, etc.) can best be added to software architectures by exercising control over the communication paths in a complex system. This conclusion was partially born out by our own Object-Oriented Database System (Open OODB) [16] work on sentries, later similar work at Object Management Group (OMG) on interceptors for adding security to systems, complex ACME connectors replacing simpler ones, and our Object Services and Consulting (OBJS) work with MCC on the Object Infrastructure Project. If Msg*Log can significantly improve the survivability of a complex agent infrastructure by exercising such control, it would serve as an important data point in this larger architectural quest.

Biographies

Tom Bannon is a Senior Member of the technical staff at Object Services and Consulting Inc. (OBJS). He is currently working on a survivability project, where OBJS is developing technology to allow graceful reorganization and restoration of threatened or failed pieces of distributed object systems. It can be described as fault tolerance growing into the distributed object world (such as engendered with CORBA), with dynamic resource, situation, threat, and failure models.

Tom has a strong interest in visual design, and specifically 3D. Tom works in his spare time on Metal Sculpture.

Prior to OBJS, Tom worked in various corporate research labs at Texas Instruments over the last 8 years. His main projects were in the areas of object-oriented databases, virtual reality game systems, people tracking vision and mapping systems, and video codecs (H.263 and MPEG4). Before that, he worked for 2.5 years in the defense area of TI, primarily building a database system for a board-level ECAD system. Immediately prior to joining OBJS he spent 6 months as a contractor to a large telecom company building web and Java-based GUIs for a network management testing system.

Tom received a B.S. in Electrical Engineering from Texas A&M University in 1984, and a M.S. in Computer Science from The University of Texas at Dallas in 1991.

At OBJS, **Steve Ford** researches, develops, and administers software and hardware. He is currently working on eGents, an agent system that communicates over E-mail. Past projects at OBJS include developing techniques to improve the Survivability of CORBA-based systems and a survey of programming languages and environments for development of Internet-based applications.

Prior to joining OBJS, Steve worked on the DARPA Open OODB and its predecessor projects at Texas Instruments' Computer Science Laboratory. He was involved in the design and implementation of virtually all components of those systems, but was primarily interested in issues related to programming language API, storage and memory management, communications, object representation and translation, namespace management, performance, and the application of OODBs to image understanding environments.

Before that, Steve developed system software for the TI Explorer Lisp Machine from 1983 to 1988. His primary duties were fixing bugs and fleshing out Richard Stallman's Lisp Machine code.

Prior work includes the first Common Lisp implementation for the Sun workstation and a few years of business application development in both commercial and academic environments.

Steve received an M.S. in Computer Science and a B.A in Biological Sciences from Indiana University.

Dr. Craig Thompson is the President of Object Services and Consulting, Inc. Craig's background spans compositional middleware architectures, federation, web and object integration, virtual office and virtual enterprise, annotations, query architectures, object database systems, and natural language interfaces.

He taught database and AI at the University of Tennessee, Knoxville, from 1977 to 1981. He joined the Central Research Laboratory at Texas Instruments (TI) in 1981 and was elected Senior Member of Technical Staff in 1985. At TI, he co-invented menu-based natural language interfaces, deployed in the DARPA/USN FRESH program in the mid-1980s; productized an extensible object-relational DBMS on the TI Explorer Lisp Machine in 1985; led TI research projects in hypermedia and engineering databases in 1987-89.

From 1990 to 1995 Craig was program manager and co-principal investigator on the DARPA Open Object-Oriented Database System

Continued on page 14

Continued from page 13

(Open OODB). The project was an important influence on the computing industry move toward component software, influential in leading to the Object Management Architecture (OMA) of the Object Management Group (OMG).

In 1995 Craig Thompson, David Wells, and Steve Ford founded Object Services and Consulting, Inc. (OBJS). Craig won and served as principal investigator on the DARPA contract Scaling Object Services Architectures to the Internet (1995-1998) and Agility: Agent -Ility Architecture (1998-2002). Thompson provided architectural consulting and review for the DARPA TRP National Industrial Information Infrastructure Protocols (NIIIP) Consortium (1995-1997), the MCC Object Infrastructure Project (OIP) (1997), and the DARPA ISO Advanced Information Technology Services (AITS) Architecture (1997-1998). He is organizer and co-chair of the OMG Internet Platform Special Interest Group, chartered to merge the OMG OMA architecture with Internet and Web standards to enable large-scale Internet-enabled object-based distributed computing, and also the OMG Agent Platform Special Interest Group, chartered to meld distributed object and multi-agent systems.

Craig holds six patents, is an IEEE Senior Member, has published over 30 papers in journals and conferences. He is a nationally recognized leader in object technology standards. He co-organized the Application Integration Architectures Workshop (1993) which convened principals from over 20 standards groups on the theme of standards convergence for enterprise computing. He also co-organized the Joint W3C/OMG Workshop on Distributed Objects and Mobile Code (1996) and organized the OMG-DARPA Workshop on Compositional Software Architectures (1998).

Craig received a B.A. in Mathematics from Stanford University in 1971. He received an M.A., and a Ph.D. in Computer Science from The University of Texas at Austin, in 1977 and 1984 respectively.

David Wells is the OBJS Vice President, and head of software research.

He was an Assistant Professor in the Computer Science Department at Southern Methodist University from 1980 to 1986 where he conducted research in databases, computer security, and computer graphics. He joined Texas Instruments in 1986 and until 1995 was principal software architect and Co-Principal Investigator for the DARPA Open OODB, which was deployed at 25 government approved alpha sites and is currently licensable as a research product from TI. Wells developed the open systems principles and mechanisms that make Open OODB extensible by third parties. The Open OODB meta architecture served as one of the bases for the Object Management Group's (OMG) Object Services Architecture, which has been adopted by over 500 computer vendors and object-oriented software companies as the basis for developing interoperable systems. David is experienced in leading large research and development projects, having been PI of four research contracts with total funding of $10.7M. He holds three patents and two patent applications in the areas of database software techniques and data security, has authored five journal articles, twelve conference proceedings, and twenty-four invited external presentations and panels. He is reviewer for various journals and conferences including the International Conferences on Very Large Databases, the VLDB Journal, ACM Transactions on Databases, IEEE Computer, IEEE Transactions on Software Engineering, and SIGMOD Conferences.

David received his D. Eng. degree in Computer Science from the University of Wisconsin-Milwaukee in 1980.

References

[1] *Joint Vision 2020*, US Department of Defense, Staff, J-7, Joint Doctrine Division Support Group, info@jointdoctrine.com, http://www.dtic.mil/jv2020/

[2] *White Paper: Application-Driven Ultra*Log Robustness Metrics*, Wells, D.L., OBJS, April, 2001, http://www.objs.com/msglog/.

[3] *Msg*Log Message Transport Policy*, Thompson, C.W and Wells, D.W., OBJS, April, 2001, http://www.objs.com/msglog/.

[4] *Estimating Service Failure*, Wells, D.L. and Langworthy, D.E., OBJS, May, 1999, http://www.objs.com/Survivability/EstimatingServiceFailure.html.

[5] *Advanced Logistics Project Homepage*, Defense Advanced Research Projects Agency, http://www.darpa.mil/iso2/alp/.

[6] *Ultra*Log Homepage*, Defense Advanced Research Projects Agency, http://www.darpa.mil/tto/programs/ultraLog.html.

[7] *Cougaar Homepage*, http://www.cougaar.net/.

[8] *Msg*Log Homepage*, Object Services and Consulting, http://www.objs.com/ultralog/.

[9] *Routing in Telecommunications Networks with "Smart" Ant-Like Agents*, Bonabeau, E., Henaux, F., Gu'erin, S., Snyers, D, Kuntz, P. and Theraulaz, G., Santa Fe Institute, 1998, http://www.santafe.edu/sfi/publications/Abstracts/98-01-003abs.html.

[10] *Horus, a Flexible Group Communication System*, Renesse, R., Birman, K. and Maffeis, S., *Communications of the ACM*, April 1996.

[11] *Advanced Logistics Project: Program Briefing*, Carrico, T., DARPA ISO, 1998, http://www.darpa.mil/iso2/alp/public_access/pr_brf_4398.

[12] *Specifying and Measuring Quality of Service in Distributed Object Systems*, Loyall, J.P., Schantz, R.E., Zinky, J.A., Bakken D.E., Proceedings of the First International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC '98), 20-22 April 1998, Kyoto, Japan, http://www.dist-systems.bbn.com/papers/1998/ISORC.

[13] *Quality of Service Adaptation for Ultra*Log Core*, Zinky, J., Ultra*Log Developers Meeting, April 2001, http://www.ultralog.net/products/meetings/4April/4l_QoSUltraLogDev-Apr2001-5.ppt.

[14] *Msg*Log Metrics*, Wells, D.W., OBJS, April, 2001, http://www.objs.com/msglog/.

[15] *DARPA Control of Agent Based Systems (CoABS) Program: The CoABS Grid*, Global InfoTek, Inc., March, 2000, http://coabs.globalinfotek.com/coabs_public/coabs_pdf/gridvision.pdf.

[16] *Architecture of an Open Object-Oriented Database Management System*, Wells, D.L., Blakeley, J., Thompson, C.W., *IEEE Computer*, Vol. 25, No. 10, October 1992.

Contact Information

Tom Bannon
Object Services and Consulting Inc.
5111 Purdue Ave.
Dallas, TX 75209

Phone: 214-902-8368
E-mail: bannon@objs.com

Steve Ford
Object Services and Consulting Inc.
225 State Rd.
West Grove, PA 19390

Phone: 610-345-0290
E-mail: ford@objs.com

Dr. Craig Thompson
Object Services and Consulting Inc.
2725 Deep Valley Trail
Plano, TX 75023

Phone: 972-612-6998
E-mail: thompson@objs.com

David Wells
Object Services and Consulting Inc.
6111 Baywood Ave.
Baltimore, MD 21209

Phone: 410-318-8938
E-mail: wells@objs.com

Object Services and Consulting Inc.

http://www.objs.com/

The Need and Tools to Gain Precision in Electronic Commerce

Gio Wiederhold, Stanford University

1. Introduction

The web is moving our decision-making processes from an information-sparse setting into in information-rich setting. A major problem facing individual users is the ubiquity and diversity of information. The World-Wide Web contains more alternatives than can be investigated in depth. The value system itself is changing, whereas traditionally information had value, now it is the attention of the purchaser that has value. Evidence is the focus on free availability of web resources and the difficulties in maintaining copyrights.

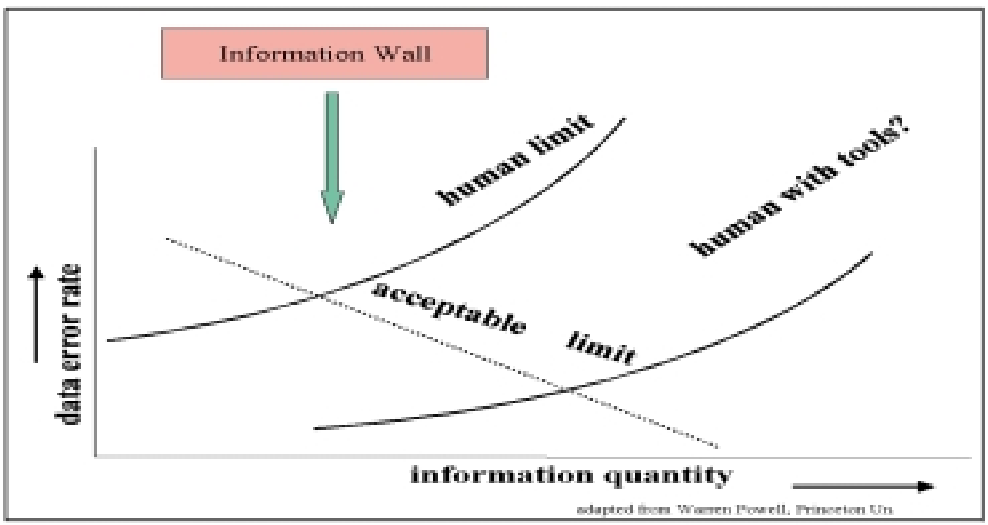

Figure 1: Limits on Human Processing

Tools are needed to search through the mass of potential information. Traditional information retrieval tools have focused on returning as much possible relevant information, in the process lowering the precision, since much irrelevant material is returned as well. However, for e-commerce to be effective, we cannot afford to wade through excess data produced by these errors. In statistics these are characterized as false positives, or type 2 errors. In most business situations, a modest fraction of missed opportunities (type 1 errors) are acceptable. We will discuss the trade-offs and present current and future tools to enhance precision in electronic information gathering.

While much progress in Information Science is triggered by progress in technology, when assessing the future we must focus on the consumers. In this paper we consider the needs of decision-makers, consumer, business, and professional needs were also addressed in an earlier source report [29]. Support of decision-making must be done expeditiously, with a very low rate of error and modest human supervision. When excessive information must be processed, efficiency drops, and what is worse, confusion ensues, making decision-making error prone.

The major problem facing individual decision-makers is the ubiquity and diversity of information. Even more than the advertising section of a daily newspaper the World-Wide Web contains more alternatives than can be investigated in depth. When leafing through advertisements the selection is based on the prominence of the advertisement, the convenience of getting to the advertised merchandise in one's neighborhood, the reputation of quality, personal or created by marketing, of the vendor, and features - suitability for a specific need, and price. The dominating factor differs based on the merchandise. Similar factors apply to on-line purchasing of merchandise and services. Lacking the convenience of leafing through the newspaper, greater dependence for selection is based on selection tools.

1.1 Getting the Right Information

Getting complete information is a question of breadth. In traditional measures completeness of coverage is termed *recall*. To achieve a high recall rapidly all possibly relevant sources have to be accessed. Since complete access for every information request is not feasible, information systems depend on having indexes. Having an index means that an actual information request can start from a manageable list, with points to locations and pages containing the actual information.

The effort to index all publicly available information is immense. Comprehensive indexing is limited due to the size of the web itself, and the rate of change of updates to the information on the web. Some of these problems can be, and are being addressed by brute force, using heavyweight indexing engines and smart indexing engines. For instance, sites that have been determined to change frequently will be visited by the *worms* that collect data from the sources more often, so that the average information is as little out-of-date as feasible [17]. Of course, sites change very frequently, say more than once a day, cannot be effectively indexed by a broad-based search engine. We have summarized the approaches currently being used in [30].

Getting complete information typically reduces the fraction of actual relevant material in the retrieved collection. It is here where it is crucial to make improvements, since we expect that the recall volume of possibly relevant retrieved information will grow as the web and retrieval capabilities grow. Selecting a workable quantity that is of greatest benefit to a customer requires additional work. This work can be aided by the sources, through better descriptive information or by intermediate services, that provide filtering. If it is not performed, the customer has a heavy burden in processing the overload, and is likely to give up.

High quality indexes can help immensely. Input for indexes can be produced by the information supplier, but those are likely to be limited. Schemes requiring cooperation of the sources have been proposed [8]. Since producing an index is a valued-added service, it is best handled by independent companies, who can distinguish themselves, by comprehensiveness versus specialization, currency, convenience of use, and cost. Those companies can also use tools that break through access barriers in order to better serve their population.

1.2 The Need for Precision

Our information environment has changed in recent years. In the past, Say ten years ago, most decision makers operated in settings where information was scarce, and there was a inducement to obtain more information. Having more information was seen as being able to make better decisions, and reduce risks, save resources, and reduces losses.

Today we have access to an excess of information. The search engines will typically retrieve more than a requestor can afford to read. The metrics for information systems have been traditionally recall and precision. Recall is defined as the ratio of relevant records retrieved to all relevant records in the database. Its complement, the count of relevant records not retrieved is termed a type 1 error in statistics. Precision is defined similarly as the ratio to relevant records to irrelevant records. The irrelevant records retrieved are categorized as type 2 errors. In practical systems these are related, as shown in Figure 2. While

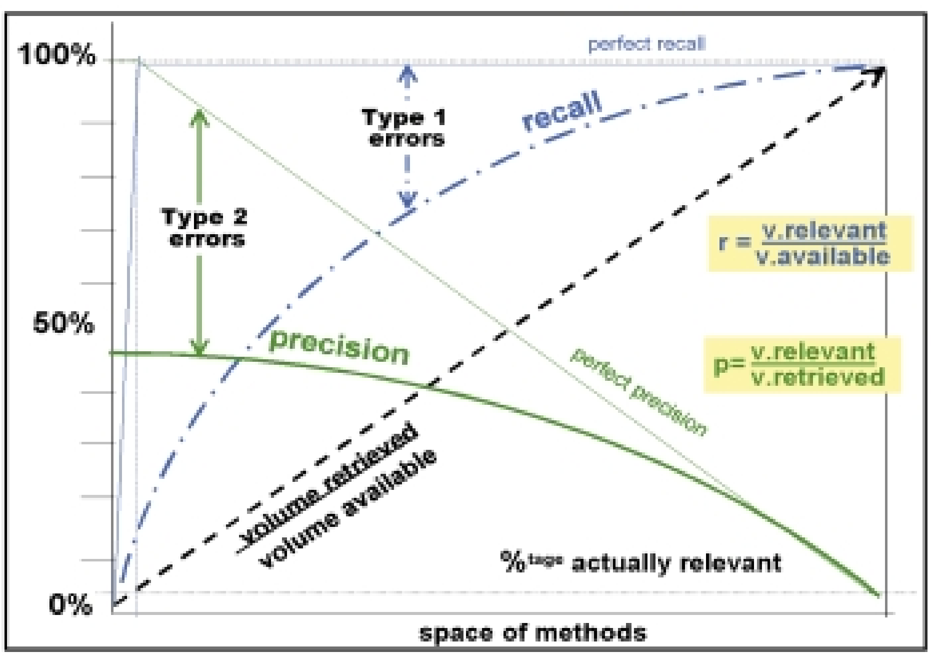

Figure 2: Trading of Relevance versus Precision

Continued on page 18

Continued from page 17

recall can be improved by retrieving more records, the precision becomes disproportionally worse.

There are a number of problems with these metrics: measuring relevance and precision, the relative cost of the associated errors, and the scale effect of very large collections.

Relevance - Well recognized is that the decision on relevance of documents is fluid. When the resources, as on the web, are immense, the designation of relevance itself can become irrelevant. Some documents add so little information that an actual decision-making process will not be materially affected. A duplicate document might be rated relevant, although it provides no new information. Most experiments are evaluated by using expert panels to rate the relevance of modest document collections, since assessing all documents in the collection is a tedious task.

Precision - The measurement of precision suffers from the same problem, although it does not require that all documents in the collection be assessed, only the ones that have actually be retrieved. Search engines, in order to assist the user, typically try to rank retrieved items in order of relevance. Most users will only look at the 10 top-ranked items. The ranking computation differs by search engine, and account for much of the differences among them. Two common techniques are aggregations of relative word frequencies in documents for the search terms and popularity of webpages, as indicated by access counts or references from peer pages [Google ref]. For e-commerce, where the catalog entries are short and references harder to collect these rankings do not apply directly. Other services, as MySimon, and Epinion [6] try to fill that void by letting users vote.

The cost of an individual type 2 error is borne by the decision-maker, who has to decide that an erroneous, irrelevant supplier was selected, perhaps a maker of toy trucks when real trucks were needed. The cost of an individual rejection may be small, but when we deal with large collections, the costs can become substantial. We will argue that more automation is needed here, since manual rejection inhibits automation.

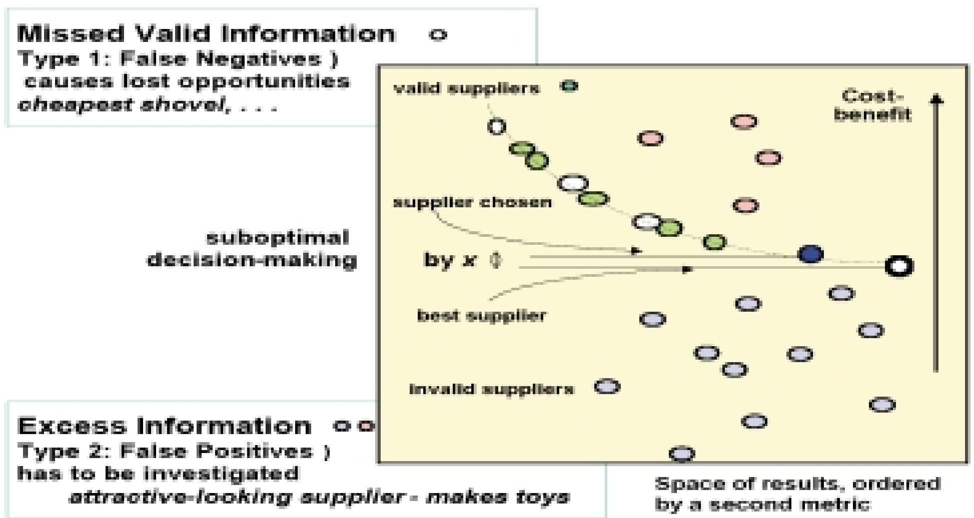

Figure 3: Costs of type 1 versus type 2 Errors

Cost - Not considered in most assessments of retrieval performance are relative costs to an actual user of the types of errors encountered. For instance, in a purchasing situation, the cost of not retrieving all the possible suppliers of an item may cause paying more than necessary. However, once the number of suppliers is such that a reasonable choice exists, the chance that other suppliers will offer significantly lower prices is small. The cost of type 1 errors is then low, as shown in Figure 3.

Scale. Perfection in retrieval is hard to achieve. In selected areas we find now precision ratios of 94% [19]. While we don't want to belittle such achievements, having 6% type 2 errors can still lead to very many irrelevant instances, when such techniques are applied to large collections, for instance, a 6% error rate on a million potential items will generate 60 000 errors, way too many to check manually. It is hard to be sure that no useful items have been missed if one restricts oneself to the 10 top-ranked items.

1.3 Errors

The reasons for having errors are manifold. There are misspellings, there is intentional manipulation of webpages to make them rank high, there is useful information that has not been accessed recently by search engines, and there are suppliers that intentionally do not display their wares on the web, because they want to be judged by other metrics, say quality, than the dominant metric when purchasing, namely price. All these sources of errors warrant investigation, but we will focus here on a specific problem, namely semantic inconsistency.

The importance of errors is also domain-dependent. A database which is perfectly adequate for one application may have an excessive error rate when used for another purpose. For instance, a payroll might have too many errors in the employee's address field to be useful for mailout. It's primary purpose is not affected by such errors, since most deposits are directly transferred to banks, and the address is mainly used to determine tax deduction requirements for local and state governments. To assure adequate precision of results when using data collected for another objective some content quality analysis is needed prior to making commitments.

2. Semantic Inconsistency

The semantic problem faced by systems using broad-based collections of information is the impossibility of having wide agreements on the meaning of terms among organizations that are independent of each other. We denote the set of terms and their relationships, following current usage in Artificial Intelligence, as an *ontology*. In our work we define ontologies in a grounded fashion, namely:

Ontology:	a set of terms and their relationships
Term:	a reference to real-world and abstract objects
Relationship:	a named and typed set of links between objects
Reference:	a label that names objects
Abstract Object:	a concept which refers to other objects
Real-world object:	an entity instance with a physical manifestation

Grounding the definitions so that they can refer to actual collections, as represented in databases, allows validation of the research we are undertaking [27]. Many precursors of ontologies have existed for a long time. Schemas, as used in databases, are simple, consistent, intermediate-level ontologies. Foreign keys relating table headings in database schemas imply structural relationships. Included in more comprehensive ontologies are the values that variables can assume; of particular significance are codes for enumerated values used in data-processing. Names of states, counties, etc. are routinely encoded. When such terms are used in a database the values in a schema column are constrained, providing another example of a structural relationship. There are thousands of such lists, often maintained by domain specialists. Other ontologies are being created now within DTD definitions for the eXtended Markup Language (XML) [5].

2.1 Sources of Ontologies

Although the term ontology is just now getting widespread acceptance, all of us have encountered ontologies in various forms. Often terms used in paper systems have been reused in computer-based systems:

Lexicon:	collection of terms used in information systems
Taxonomy:	categorization or a classification of terms
Database schemas:	attributes, ranges, constraints
Data dictionaries:	guide to systems with multiple files, owners
Object libraries:	grouped attributes, inherit., methods
Symbol tables:	terms bound to implemented programs
Domain models:	interchange terms in XML DTDs, schemas.

The ordering in this list implies an ongoing formalization of knowledge about the data being referenced.

Continued on page 20

Continued from page 19

Database schemas are the primary means used in automation to formalize ontological information, but they rarely record relationship information, nor define the permissible range for data attributes. Such information is often obtained during design, but rarely kept and even less frequently maintained. Discovering the knowledge that is implicit in the web itself is a challenging task [9].

2.2 Large versus Small Ontologies

Of concern is the breadth of ontologies. While having a consistent, world-wide ontology over all the terms we use would cause the problem of semantic inconsistency to go away, we will argue that such a goal is not achievable, and, in fact, not even desirable.

Small Ontologies

We have seen successes with small, focused ontologies. Here we consider groups of individuals, that cooperate with some shared objective, on a regular basis. Databases within companies or interest groups have been effective means of sharing information. Since they are finite, it is also possible for participants to inspect their contents and validate that the individual expectations and the information resources match. Once this semantic match is achieved, effective automatic processing of the information can take place. Many of the ongoing developments in defining XML DTD's and schemas follow the same paradigm, while interchanging information to widely distributed participants. Examples are found in diverse applications, as petroleum trading and the analysis of Shakespeare's plays. The participants in those enterprises have shared knowledge for a long time, and a formal and processable encoding is of great benefit.

There is still a need in many of these domains to maintain the ontologies. In healthcare, for instance, the terms needed for reporting patient's diseases to receive financial reimbursement change periodically, as therapies evolve and split for alternate manifestations. At a finer granularity, disease descriptors used in research areas evolve even faster, as we learn about distinctions in genotypes that affect susceptibility to diseases.

The maintenance of these domain ontologies often evolves onto professional associations. Such associations have a membership that has an interest in sharing and cooperating. Ontology creation and maintenance is a natural outgrowth of their function in dissemination of information, and merges well with the role they have in publication and organizing meetings. An example of such a close relationship in Computer Science is the classification of computer literature [1], published by the ACM and revised approximately every 5 years. This document provides an effective high-level view to the literature in the scientific aspects of the domain, although it does not provide a granularity suitable for, say, trading and purchasing of software.

Large Ontologies

A major effort, sponsored by the National Library of Medicine (NLM), has integrated diverse ontologies used in healthcare into the Unified Medical Language System (UMLS) [11]. In large ontologies collected from diverse sources or constructed by multiple individuals over a long time some inconsistencies are bound to remain. Maintenance of such ontologies is required when sources change [21]. It took several years for UMLS to adapt to an update in one of its sources, the disease registry mentioned earlier. Still. UMLS fulfills is mission in broadening searches and increasing recall, the main objective of bibliographic systems.

Large ontologies have also been collected with the objective to assist in common-sense reasoning (CyC) [16]. CyC provides the concept of *microtheories* to circumscribe contexts within its ontology. CyC has been used to *articulate* relevant information from distinct sources without constraints imposed by microtheories [4]. That approach provides valuable matches, and improves recall, but does not improve precision.

The inconsistency of semantics among sources is due to their autonomy. Each source develops its terminology in its own context, and uses terms and classifications that are natural to its creators and owners. The problem with articulation by matching terms from diverse sources is not just that of synonyms - two words for the same object, or one word for completely different objects, as *miter* in carpentry and in religion. The inconsistencies are much more complex, and include overlapping classes, subsets, partial supersets, and the like. Examples of problems abound. The term *vehicle* is used differently in the transportation code, in police agencies, and in the building code, although over 90% of the instances are the same.

The problems of maintaining consistency in large ontologies is recursive. Terms do not only refer to real-world objects, but also to abstract groupings. The term 'vehicle' is different for architects, when designing garage space, versus its use in traffic regulation, dealing with right-of-way rules at intersections. At the next higher level, talking about transportation will have very different coverage for the relevant government department versus a global company shipping its goods.

There are also differences in granularity with domains. A vendor site oriented towards carpenters will use very specific terms, say *sinkers* and *brads*, to denote certain types of nails, that will not be familiar to the general population. A site oriented to home owners will just use the general categorical term nails, and may then describe the diameter, length, type of head, and material. For the homeowner to share the ontologies of all the professions involved in construction would be impossible. For the carpenter to give up specialized terms and abbreviations, as 3D for a three-penny sized nail, would be inefficient — language in any domain is enhanced to provide effective communication within that domain. The homeowner cannot afford to learn the thousands of specialized terms needed to maintain one's house, and the carpenter cannot afford wasting time by circumscribing each nail, screw, and tool with precise attributes.

The net effect of these problems, when extended over all the topics we wish to communicate about is that **it is impossible to achieve a globally consistent ontology**. Even if such a goal could be achieved, it could not be maintained, since definitions within the subdomains will, and must continue to evolve. It would also be inefficient, since the subdomains would be restricted in their use of terms. The benefits to the common good, that we all could communicate consistently will be outweighed by the costs incurred locally and the cost of the requirements that we all acquire consistent global knowledge.

Composition of Small Ontologies

If we have proven here, albeit informally, that large global ontologies cannot be achieved, even though they are desirable to solve broader problems than can be solved with small ontologies, we are faced with one conclusion. It will be necessary to address larger problems we interoperating with small ontologies. Since a simple integration of small ontologies will lead us directly into the problems faced by large ontologies, we must learn to combine the small ontologies as needed, specifically as needed for the applications that require the combined knowledge.

However, inconsistent use of terms makes sharing of information from multiple sources incomplete and imprecise. As shown above, forcing every category of customers to use matching terms is inefficient. Mismatches are rife when dealing with geographic information, although localities are a prime criterion for articulation [18].

Most ontologies have associated textual definitions, but those are rarely sufficiently precise to allow a formal understanding without human interpretation. Although these definitions will help readers knowledgeable about the domain, they cannot guarantee precise automatic matching in a broader context, because the terms used in the definitions also come from their own source domains. The result is that inconsistencies will occur when terms for independent, but relatable domains are matched.

These inconsistencies are a major source for errors and imprecision. We have all experienced web searches that retrieved entries that had identically spelled keywords, but were not all related to the domain we are addressing - type 2 errors. When we augment the

Continued on page 22

Continued from page 21

queries with possible synonyms, because we sense a high rate of missing information, type 1 errors, the fraction of junk, type 2 errors, typically increases disproportionably. The problems due to inconsistency are more of a hindrance to automation than to browsing, where one deals with one instance at a time.

3. Articulation

Since we cannot hope to achieve global consistency, but still must serve applications that span multiple domains, we must settle composition. The theme, that only focused, application oriented approaches will be maintainable, directs us to limit us to the concepts needed for interoperation, for which we will reuse the term articulation.

Once we have clear domain ontologies that are to be related within an application we must recognize their intersections, where concepts belong to multiple domains. For clarity, we restrict ourselves to intersections of two domains. More complex cases are certainly feasible, but we will address them using the algebraic capabilities presented in Section 5. We will deal in this section with the binary case.

3.1 Semantic Rules

An application requiring information from two domains must be able to join them semantically, so that there will be a semantic intersection between them. Such a match may not be found by lexical word matching. For instance, checking for a relationship of automobile purchasing and accidents requires looking for the car *owners* in dealer records that list the *buyers*.

We define then the articulation to be the semantically meaningful intersection of concepts that relate domains with respect to an application. The instances should match according to our definition of an ontology, given in the introduction to this section.

An articulation point hence defines a relevant semantic match, even if the actual terms and their representation do not match. For instance, for vacation travel planning a *trip segment* matches the term *flight* from the airline domain, and term *journey* from the railroad domain. Terms at a lower level of abstraction, defining instances also have to made to match. For instance, to take a train to San Francisco Airport one must get off at the San Bruno Caltrain station. Here the terms are at the same granularity, and once matched, the articulation is easy. Understanding such articulation points is a service implicitly provided now by experts, here travel agents. In any application where subtasks cross the boundaries of domain some experts exist that help bridge the semantic gaps.

Often the matching rules become complex. In listings of the California Department of Motor Vehicles (DMV) houseboats are included. To match vehicles correctly for, say, an analysis of fuel consumption, the articulation rule has to exclude those houseboats. The attributes that define the classes now become part of the input needed for the execution of the articulation. Such differences in scope are common, and yet often surprising, because the application designer has no reason to suspect that such differences exist. A good way to check correctness of matches is to process the underlying databases.

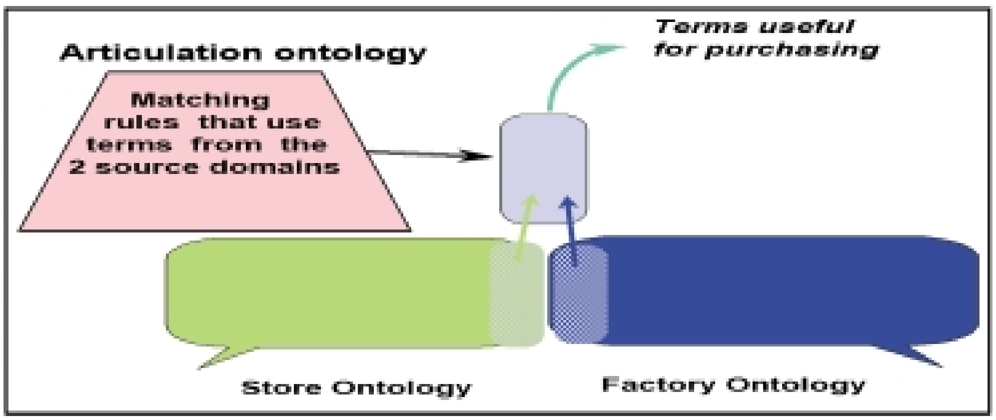

Figure 4: Articulation of Two Domains

The concept is not to force alignment of entire base ontologies, but only present to the application consistent terms in the limited overlapping area. Typical applications that rely on intersections are in purchasing goods and services from another domain, the example above cited *journeys* and *lights*. Terms only used in one domain need not be aligned, as *sleeping compartment* and *in-flight movie*.

3.2 Creating Articulations

There are already people in all kinds of business settings who perform such work. Any travel agent has to be able to deal with the diversity of resources. However, when interacting the phone or directly with diverse webpages on the Internet, the problems are not widely recognized. For automation they will need to be solved formally.

Keeping the rules that define an articulation specific to narrow application contexts simplifies their creation and maintenance. Even within an application area multiple rule sets can exist, for instance one might specific to logistics in drug distribution. The logical organization to be responsible for the rules which define such a specific *articulation ontology* for, say, pharmaceutical drugs would be the National Drug Distributors Association (NDDA) in the USA. There will be a need for tools to manage those rules, and these tools can serve diverse applications, both in creation and maintenance [13].

When two sources come from the same organization, we would expect an easy match. i.e., a consistent ontology. However, we found that even in one company the payroll department defined the term *employee* differently from the definition used in *personnel*, so that the intersection of their two databases is smaller than either source. Such aberrations can easily be demonstrated, by computing the differences of the membership from the respective databases, following an ontological grounding as we use here. In large multi-national corporations and in enterprises that have grown through mergers, differences are bound to exist. These can be dealt with if the problems are formally recognized and articulated, but often they are handled in an isolated fashion, and solved over and over in an ad-hoc fashion.

Such analyses are not feasible when source information sources are world-wide, and contexts become unclear. Here no comprehensive matching can be expected, so that certain operation cannot be executed reliably on-line, although many tasks can be carried out. These difficulties are related to the applicability of the closed-world-assumption (CWA) [22].

It requires an effort to define articulations precisely. The investment pays off as it reduces the wasted effort in taking care of the effects of errors that are now avoided. The initial effort becomes essential to support repetitive transactions, where one cannot afford to spend human efforts to correct semantic mismatches every time.

To summarize, articulations that are needed among domains are made implicitly by *smart* people. Converting human expertise in dealing with domain intersections to permit automation will require a formalization of the domain ontologies and their semantic intersections. Such research will be an important component of moving to the semantic web [2]

3.3 An Algebra for Ontologies

There will be many applications that require more than a pair of ontologies. For example, logistics, which must deal with shipping merchandise via a variety of carriers: truck, rail, ship, and air, requires interoperation among many diverse domains, as well as multiple companies located in different countries. To resolve these issues we are developing an ontology algebra, which further exploits the capabilities of rule-based articulation [20].

Once we define an intersection of ontologies through articulation, we should also define union and difference operations over ontologies [24]. We apply the same semantic matching rules we used for articulation to transform the traditional set operations to operations that are cognizant of inter-domain semantics. Assuring

Continued on page 24

Continued from page 23

that soundness and consistency, mirroring what we expect from traditional set operations, is a challenge.

Having an algebra not only achieves disciplined scalability to an unlimited set of sources, but it also provides a means to enumerate alternate composition strategies, assess their performance, and, if warranted, perform optimizations [13]. We expect that the semantic union operation will mainly be employed to combine the results of prior intersections, in order to increase the breadth of ontological coverage of an application.

The semantic difference operation will allow the owners of a domain ontology to distinguish the terms that the owners can change as their needs change. The excluded terms, by definition, participate in some articulation, and changes made to them will affect interoperation with related domains, and hence make some application less precise, or even disable it. Informally, difference allows ontology owners to assess the scope of their local autonomy.

4. Architecture

We use the term architecture to refer to the composition of modules of information systems. Traditional information systems have depended on human experts. Their elimination through the capability of providing direct linkages on the web has led to disintermediation [23] .

We see these services being replaced by automated engines, positioned between the information clients and the information resources. Within the mediators will be the intelligent functions that encode the required expertise for semantic matching and filtering [15]. Composition of synergistic functions creates a mediator performing substantial service. Such a service is best envisaged as a module within the networks that link customers and resources, as sketched in Figure 5. Many customers can share mediator services through their web portals [14]. Multiple mediators will often be needed, when measures used for selection and valuation are nor commensurate. For instance, tradeoffs involving cost versus quality, or risk versus having-up-to-date information must be relegated to the decision maker, and not automated at lower levels in a system hierarchy.

Domain-type mediators can integrate domains as financial information, personnel management, travel, logistics, technology etc. [24]. Within these domains will be further specialization, as in finance to provide information about investing in precious metals, bonds, blue-chip stocks, utilities, and high tech.

There will be meta-services as well, helping to locate those services and reporting on their quality. Mediators encompass both experts and software to perform these functions, and sustain the services as functional requirements and underlying ontologies evolve.

4.1 Middleware and Mediation

The need for middleware to connect clients to servers has been well established. although it has only attracted a modest amount of academic interest [7]. However, middleware products only enable communication, and deal with issues as establishing connectivity, reliability, transmission security, and resolution of differences in representation and timing. A mediator can exploit those technologies and avoid dealing with the problems that arise from

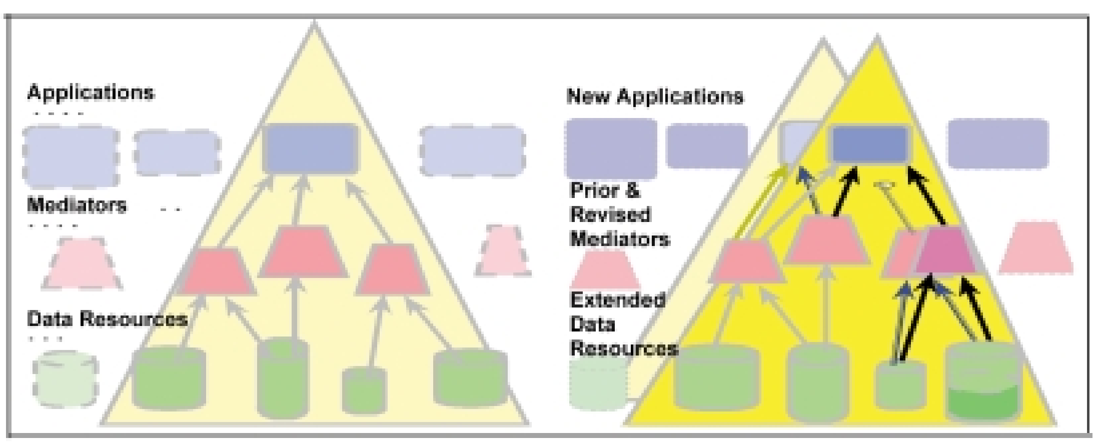

Figure 5. Mediator Architecture

having an excess of standards. However, middleware never deals with true semantic differences and only rarely with integration of information, leaving these tasks to a superior layer of software.

There is today a small number of companies building mediators with transformation and integration capability [28]. However, the available technology is not yet suitable to be shrink-wrapped and requires substantial adaptation to individual settings. Many products focus on specific increments. When the added value is modest, then the benefit gained is likely outweighed by the cost in performance incurred when adding a layer into an information system architecture. In those cases, making incremental improvements to the sources, as providing object transforms, or in applications, as providing multiple interfaces, seems to be preferable.

However, if the services placed into an intermediate layer are comprehensive, sufficient added value can be produced for the applications that access the mediators and the cost of transit through the additional layer will be offset.

4.2 Incremental Maintenance

To deliver valuable services, mediators will have to be updated as well. Some changes are bound to affect the customers, as new interfaces or changes in the underlying ontologies. Unwanted updates, scheduled by a service, often hurt a customer, even though in the long run the improvement is desired. To allow customers to schedule their adaptation to new capabilities when it is suitable for them, mediator owners can keep prior versions available. Since mediators are of modest size and do not hold voluminous data internally, keeping an earlier copy has a modest cost.

The benefits of not forcing all customers to change interfaces at the same time are significant. First of all customers can update at a time when they can do it best. A second benefit is that first only a few customers, namely those that need the new capabilities will be served. Any errors or problems in the new version can be repaired then, in cooperation with those customers, and broader and more serious problem will be avoided [25].

Since maintenance of long-lived artifacts, including software, is such a large fraction of the lifetime cost it is crucial to plan for maintenance, so that maintenance can be carried out expeditiously and economically. Being able to be responsive to maintenance needs increases consumer value and reduces both consumer and provider cost. Where maintenance today often amounts to 80% of lifetime cost, a 25% reduction in those costs can double the funds available for systems improvements, while a 25% increase can inhibit all development and lead to stasis.

5. Summary

Information presented to customers or applications must have a value that is greater than the of obtaining and managing it. A large fraction of the cost is dealing with erroneous and irrelevant data, since such processing requires human insight and knowledge. More information is hence not better, and less may well be, if relevance per unit of information produced is increased

The need for assistance in obtaining relevant information from the world-wide-web was recognized early in the web's existence [BowmanEa:94]. This field has seen rapid advances, and yet the users remain dissatisfied with the results. Complaints about information overload' abound. Web searches retrieve an excess of references, and getting an actually relevant result, requires much subsequent effort.

In this article we focused on one aspect, namely precision, the elimination of excess information.. The main method we presented is constrained and precise articulation among domains, to avoid the errors that occur when searches and integration of retrieved data is based on simply lexical matches.

We refer to the services that replace traditional human functions in information generation as mediators, and place them architecturally between the end-users, the human professionals and their local client software, and the resources, often

Continued on page 26

The Need and Tools to Gain Precision in Electronic Commerce

Continued from page 25

legacy databases and inconsistently structured web sources. Such novel software will require more powerful hardware, but we are confident that hardware-oriented research and development is progressing, and will be able to supply the needed infrastructure. The major reason for slow acceptance of innovations is not the technology itself, but the massiveness of the organizational and human infrastructure.

Biography

Dr. Wiederhold is currently a Professor Emeritus at Stanford University in both the Departments of Computer Science, and Medicine and Electrical Engineering (by courtesy) where his research includes; design and operation of large-scale software systems, database design, knowledge bases, distributed systems, applications in medicine, planning, and business, and privacy/security.

Dr. Wiederhold os also the CEO of Symmetric Security Technologies, a software development and consulting firm specializing in Internet security. He is greatly concerned about secure information management. He has been the principal investigator for NSF and DARPA-funded research projects to develop security mediation in the healthcare domain (Trusted Interoperation of Healthcare Information (TIHI)), for medical images (Trusted Image Dissemination (TID)), and CAD (Secure Access Wrapper (SAW)).

Dr. Wieferhold has authored nine books and over 300 technical reports and papers. He received his Ph.D. in Medical Information Science from the University of California in 1976.

Contact Information

Gio Wiederhold
Computer Science Department
Gates Computer Science Bldg. 4A
Room 433
Stanford University
Stanford CA, 94305

Phone: 650-725-8363
E-mail: gio@cs.stanford.edu

References

[1] Neal Coulter, et al: ACM Computing Classification System, 1999, http://www.acm.org/class/

[2] Tim Berners-Lee, Jim Hendler, and Ora Lassila: "The Semantic Web"; *Scientific American*, May 2001.

[3] C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber and Michael F. Schwartz: "The HARVEST Information Discovery and Access System"; *Proceedings of the Second International World Wide Web Conference*, Chicago, Illinois, October 1994, pp 763—771.

[4] C. Collet, M. Huhns, and W-M. Shen: "Resource Integration Using a Large Knowledge Base in CARNOT"; *IEEE Computer*, Vol.24 No.12, Dec.1991.

[5] Dan Connolly (ed.): XML: Principles, Tools, and Techniques, O'Reilly, 1997.

[6] R.V. Guha: Check Before you Buy; Epinions.com, 2000.

[7] The Garther Group: Middleware; Gartner group report, 1995.

[8] L. Gravano , H. Garcia-Molina, and A. Tomasic: "Precision and Recall of GlOSS Estimators for Database Discovery"; Parallel and Distributed Information Systems, 1994.

[9] J. Heflin, J. Hendler, and S. Luke: "Reading Between the Lines: Using SHOE to Discover Implicit Knowledge from the Web"; in AAAI-98 Workshop on AI and Information Integration, 1998.

[10] Darrel Huff: How to Lie with Statistics; Norton, 1954.

[11] Betsy Humphreys and Don Lindberg: "The UMLS project : Making the conceptual connection between users and the information they need"; Bulletin of the Medical Library Association, 1993, see also http://www.lexical.com/.

[12] Jan Jannink, Prasenjit Mitra, Erich Neuhold, Srinivasan Pichai, Rudi Studer, Gio Wiederhold: "An Algebra for Semantic Interoperation of Semistructured Data"; in 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99), Chicago, Nov. 1999.

[13] Jan Jannink: Towards Semantic Interoperation of Semistructured Data; Stanford University, Computer Science Department PhD thesis, March 2001, http://www-db.stanford.edu/~jan/.

[14] YuHui Jin, Stefan Decker, and Gio Wiederhold: OntoWebber: Model-Driven Ontology-Based Web Site Management; in preparation, Stanford University CSD, Ontoweb project, May 2001.

[15] Thomas Langer: "MeBro - A Framework for Metadata-Based Information Mediation"; First International Workshop on Practical Information Mediation and Brokering, and the Commerce of Information on the Internet, Tokyo Japan, September 1998, http://context.mit.edu/imediat98/paper2/.

[16] D. Lenat and R.V. Guha: Building Large Knowledge-Based Systems, Addison-Wesley (Reading MA), 372 pages.

[17] Clifford Lynch: "Searching the Internet"; The Internet: Fulfilling the Promise; *Scientific American*; March 1997.

[18] David Mark et al.: Geographic Information Science: Critical Issues in an Emerging Cross-Disciplinary Research Domain; NCGIA, Feb. 1999, http://www.geog.buffalo.edu/ncgia/workshop report.html.

[19] Tom Mitchell: "Machine Learning and Data Mining"; *Communications of the ACM*, Vol. 42, No. 11, November 1999.

[20] Prasenjit Mitra, Gio Wiederhold, and Martin Kerstens "A Graph-Oriented Model for Articulation of Ontology Interdependencies"; *Proceedings of Extending DataBase Technologies,* (EDBT 2000), Konstanz, Germany, Springer Verlag LNCS, March 2000;

[21] Diane E. Oliver and Yuval Shahar: "Change Management of Shared and Local Health-Care Terminologies"; IMIA Working Group 6 Meeting on Health Concept Representation and Natural Language Processing, Phoenix, AZ, 1999.

[22] R. Reiter,R.: "On Closed World Data Bases"; in *Logic and Data Bases, Gallaire and Minker* (eds) Plenum Press NY., 1978,

[23] Gio Wiederhold, Gio: "Mediators in the Architecture of Future Information Systems"; *IEEE Computer*, March 1992, pages 38-49; reprinted in Huhns and Singh: Readings in Agents; Morgan Kaufmann, October, 1997, pp.185-196.

[24] Wiederhold, Gio: "An Algebra for Ontology Composition"; *Proceedings of 1994 Monterey Workshop on Formal Methods*, Sept 1994, U.S. Naval Postgraduate School, Monterey CA, pages 56-61.

[25] Gio Wiederhold: "Modeling and System Maintenance"; in Papazoglou: OOER'95: Object-Oriented and Entity Relationship Modelling; Springer Lecture Notes in Computer Science, Vol. 1021, 1995, pp. 1-20.

[26] Gio Wiederhold: "Customer Models for Effective Presentation of Information"; Position Paper, Flanagan, Huang, Jones, Kerf (eds): Human-Centered Systems: Information, Interactivity, and Intelligence, National Science Foundation, July 1997, pp.218-221.

[27] Gio Wiederhold and Michael Genesereth: "The onceptual Basis for Mediation Services"; *IEEE Expert, Intelligent Systems and their Applications*, Vol.12 No.5, Sep-Oct.1997.

[28] Gio Wiederhold: "Weaving Data into Information"; Database Programming and Design, Freeman pubs, Sept. 1998.

[29] Gio Wiederhold: Trends in Information Technology; report to JETRO.MITI, http://www-db.stanford.edu/pub/gio/1999/miti.htm.

[30] Wiederhold, Gio: "Precision in Processing Data from Heterogeneous Resources"; in B.Lings and K.Jeffreys (eds.): Advances in Databases, Procedings of 17th British National Conference on Databases, Exeter, UK, July 2000, pages 1-18.

Acknowledgments

Background material for this article came from a study prepared for JETRO / MITI [29] , and in turn was based on many published resources as well as on discussions with wonderful people that I have encountered in my work and studies. Work on mediation in general was supported by DARPA for many years. The research focusing on articulation is being supported by AFOSR under the New World Visions initiative and by DARPA's DAML project.

STN Vol. 4, No. 4
In This Issue

Data & Analysis Center for Software
P.O. Box 1400
Rome, NY 13442-1400

Return Service Requested

PRSRT STD
U.S. Postage
PAID
Permit #566
UTICA, NY